International Journal of Computer Networking,
Wireless and Mobile Communications (IJCNWMC)
ISSN(P): 2250-1568; ISSN(E): 2278-9448
Vol. 4, Issue 2, Apr 2014, 115-124
© TJPRC Pvt. Ltd.

APPROACH TOWARDS SAFETY IN WSN: A SURVEY

PADMINI M. S & PRASANNA KUMAR G

The National Institute of Engineering, Mysore, Karnataka, India

ABSTRACT

Wireless sensor networks gaining more popularity and continue to grow in large extent and so there is need for effective security and safety mechanisms. Sensor networks operate in hostile environments and they interact with sensitive data and they do not have any user controlling the individual nodes. Wireless Sensor networks can be used in various areas like battlefields, commercial applications such as Traffic monitoring, Environment monitoring and also in buildings and smart homes and in many other scenarios. So the security concerns need to be addressed from the initial stage of the system design and it is one of the important challenges in wireless sensor networks. Compared to traditional networks, there are lot of security challenges involved in sensor networks due to existing computing and resource constraints. This provides chance for enormous research potential in wireless sensor network security. So the current research in the field of wireless sensor network will be of great benefit to the researchers. So, in this paper we have done the survey of new technologies of wireless sensor network security in different scenarios and the ways of providing security in the sensor networks, classifying different attacks and the defensive measures that can be taken in the sensors networks and also cryptographic strength and analysis of performance of these scenarios. Cloning attack will be identified using this model. Zero knowledge protocol can be applied for verifying the sender sensor nodes. Clone attack can be addressed with attachment of unique fingerprint to each node. We have used the zero knowledge protocol to address the non transmission of crucial cryptographic information in the wireless sensor network. Thus, replay attack and man-in-the middle attack can be prevented.

KEYWORDS: Crucial Cryptographic Information, Cloning Attack, Wireless Sensor Networks

INTRODUCTION

Wireless sensor nodes are nowadays going through lot of advancement in their technologies. Basically, these sensor nodes are having a variety of compact wireless sensors. Manual intervention in the nodes will be less after deploying the nodes in different environment and also in controlling the nodes. One of the common physical attacks in the sensor network is node cloning. In this, the attacker will be identifying the authorized nodes and he will be creating the copy of cryptographic information to make clones, and these nodes will be deployed into the network by using commodity hardware and operating system. In this paper we have classified various attacks in a sensor network and method implemented to prevent them. By using this method, Clone attack, reply attack and man in the middle attack can be identified. We use the zero knowledge protocol for authenticating the authorized sender.

There are different attacks that occur in WSN but certain active attacks that can be identified with our model are as follows:

Clone Attack: In clone attack, an intruder may capture a sensor node and copy the cryptographic information to another node known as cloned node. Then, this cloned sensor node can be installed to capture the information of the network. The intruder can also insert false information, or modify the information passing through cloned nodes [1]. Continuous physical monitoring of nodes is not possible to detect potential tampering and cloning. Thus an effective scheme for detection is necessary to prevent these attacks [2].

Scenario 1: When a cloned node behaves as a cluster head the cluster heads communicate with base station. The base station becomes the verifier and questions the cloned cluster head and then detects the cloning attack through ZKP.

Scenario 2: Any other existing id with same fingerprint may be used as cloned node: As a node gets compromised its clones are injected to network which tries to make communication. Only after the verification of clone nodes they are able to communicate with other nodes Figure 1 shows how node ’6’ of cluster ’2’ is cloned and placed in cluster ’1’ with a new id ’2’. Cloned node uses the finger print “s” of node ’6’; it fails to authenticate itself during communication through ZKP.

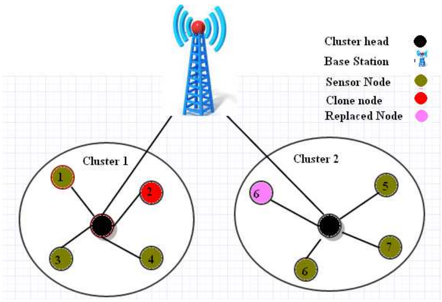

Figure 1: Security Analysis

Scenario 3: When already present id with a different finger print get used by clone nodes: The cloned node with some existing Id get detected every time by the neighboring nodes as the secret finger print of the cloned node will not match with the finger print possessed by the neighbors.

Scenario 4: When same id and same fingerprint used by cloned node: If it uses the same id ’6’, the cluster head of cluster 1 will reject any communication as node ’6’ as it is not a member of cluster ’1’. The base station which will detect immediately at the initiation of the communication request. This scenario is depicted in Figure 1.

Man in the Middle Attack

It is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them think that they are talking directly to each other over a private connection, the entire conversation is controlled by the attacker. The attacker must be able to intercept all messages going between the two victims and can insert new ones. In our model, the finger print of a node never gets transmitted and thus intruder do not have a chance to identify them. Even if the attacker tries to generate a finger print in some brute force method, it will not be able to escape the check as every time a new public key N and a new random will be used[3].

Replay Attack

A replay attack is a form of network attack in which a valid data transmission is maliciously repeated or delayed. This is carried out either by the originator or by intruder who intercepts the data and retransmits it. In this attack, an intruder tries to replay the earlier communication and authenticate itself to the verifier. But, with our model verifier will be sends different values for each communication [4].

We classify the main aspects of wireless sensor network security into three major categories:

- Attacks and defensive measures.
- The obstacles to sensor network security.
- The requirements of a secure wireless sensor network.

In cryptography, a **zero-knowledge proof** or **zero-knowledge protocol** is a method by which one party (the *prover*) can prove to another party (the *verifier*) that a given statement is true, without conveying any additional information. This is a particular case known as *zero-knowledge proof of knowledge*, for zero-knowledge proofs of knowledge; the protocol must necessarily require input from the verifier, such that the responses from the prover will convince the verifier [5].

Zero-knowledge (ZK) protocols allow A to demonstrate knowledge of a secret to B without revealing any useful information about that secret. The prover (A) is the entity that is trying to prove knowledge of a secret. The verifier (B) is the entity that verifies that the prover has knowledge of a secret. The first message, sent from A to B in a 3-pass identification protocol, is called the witness or commitment. The second message, sent from B to A, is called the challenge. The final message, sent from A to B, is called the response.

ZK protocols must satisfy 3 properties:

- **Soundness:** This property states that if the statement is false, the prover who is cheating can convince the verifier that it is true, except with some small probability.
- **Zero-Knowledge:** This property states that if the statement is true, no cheating verifier can learn anything.
- **Completeness:** This property states that if the statement is true, the verifier will be convinced by the prover.

DISADVANTAGES OF PREVIOUS METHODS

In simple password protocols, a claimant A gives his password to a verifier B. If certain precautions are not taken, an eavesdropper can get hold of the password that was transferred, and from there on he can impersonate A to his liking. Other protocols try to improve on this, as in the case of challenge-response systems. In this sort of protocols, A responds to B’s challenge to prove knowledge of a shared secret. Of course, the challenge is changed every time the protocol is used; therefore, an eavesdropper can gain partial information about the shared secret key. Zero Knowledge Protocols (ZKP) which are designed to defeat the disadvantages described above. In ZKP, a prover will try to demonstrate knowledge of a certain secret to a verifier. The main idea is to allow the proof to take place without revealing any information about the proof itself, except of course for the fact that it is indeed a valid one [5].

ZERO KNOWLEDGE PROTOCOL

Zero Knowledge Protocols, is an improvement on these situations. The objective is to obtain a system in which it is possible for a prover to convince a verifier of his knowledge of a certain secret without disclosing any information. The present invention relates to Zero Knowledge Protocols that allows the knowledge of some “secret” or private key information in a first party domain to be verified by a second party without imparting the actual secret information or private key to that second party or to any eavesdropping third party. The secret information may be any numeric value, hereafter referred to as the secret number of the prover P. ZKP based protocols require less bandwidth, less computational power, and less memory compared to other authentication methods and thus seems to be suitable for WSN.

ADVANTAGES OF ZERO KNOWLEDGE PROTOCOL

Zero Knowledge Protocols have the following properties:

- The verifier cannot learn anything from the protocol. This is the central concept of zero knowledge, i.e., zero amount of knowledge is transferred.
- The prover cannot cheat the verifier.
- The verifier cannot pretend to be the prover to a third party.

Efficiency: The computational efficiency of ZK protocol is because of its interactive proofs nature. The costly computation related to encryption is avoided.

Degradation: The security of protocol itself does not get degraded with continuous use as no information about the secret is divulged.

Unsolved Mathematical Assumptions: ZK protocols are based on various mathematical Problems like integer factorization and discrete algorithms.

Three Stage ZK Protocols

A large class of zero-knowledge protocols consists in repeating *n* times the following three message rounds:

- Claimant (Prover) to Verifier: Witness
- Verifier to Claimant: Challenge
- Claimant to Verifier: Response

The prover selects a random element from a pre-defined set as its secret commitment and from this computes a public witness. The randomness creates unrepeatable execution histories. The claimant basically asserts that it can answer a number of questions. The verifier probabilistically tests this by asking one of these questions. If the claimant is the one it claims to be, then it can answer all questions successfully. The answer to any one of these questions does not provide information about the secret commitment. The Verifier checks the answer for accuracy. The protocol is repeated *n* times.

Protocols

The three protocols that will be covered are Fiat-Shamir, Guillou-Quisquarter(GQ), and Schnorr protocols. Each protocol has its advantages and disadvantages. After the descriptions of the protocols are given, a comparison between them will be discussed.

Fiat-Shamir Identification Protocol

The **Fiat–Shamir** is a technique in cryptography for taking an interactive proof of knowledge and creating a digital signature based on it. The original interactive proof must have the property of being public-coin, for the method to work. In cryptography, ZKP’s are primarily used as a means of entity authentication. That is alice possesses some secret S that only she can know. She proves to bob that she is indeed alice(and not an impostor) by proving that she possesses S.

of course, she wants to do so without revealing S to bob (or any potential eavesdroppers). The Fiat-Shamir heuristic was originally presented without a proof of security; later, Point cheval and Stern proved its security against chosen message attacks in the *random oracle model*, that is, under the assumption that random oracles exist. In the case that random oracles don't exist, the Fiat–Shamir heuristic has been proven insecure by Gold wasser and Kalai. The Fiat–Shamir heuristic thus demonstrates a major application of random oracles[6].

Fiat-Shamir identification protocol is an example of ZK protocol. In this protocol Alice proves to Bob her knowledge of a secret, S using many rounds of three message challenge-responses. This protocol is more suited to smart card applications, as it tries to keep the size of each accreditation (i.e., each round) to a minimum. But it does require about 3 times the computational power of the Fiat-Shamir protocol.

Step 1: A random modulus, *n*, product of two large prime numbers *p* and *q*, is generated by a Trusted Party. The trusted party keeps the primes *p* and *q* secret and publishes *n*.

Step 2: Alice, the prover selects a secret *s*, relatively prime to *n*. Alice, then makes v ($=s^2$) public.

Step 3: To prove her knowledge of the secret *s*, Alice chooses a random number r, ($1 \leq r \leq n-1$) using a random generator. She sends $x = r^2$ mod *n*, to Bob, the verifier. This is her commitment to authentication.

Step 4: Bob randomly sends either a 0 or a 1 as *e*, his challenge.

Step 5: Alice computes the response $y = r\ s^e$ mod *n*, where $e \in \{0,1\}$ is the challenge she receives from Bob. Thus, depending on Bob's challenge, 0 or 1, Alice responds with *r* or, $r.s^e$ mod *n*.

Step 6: Bob accepts the response upon checking $y^2 \equiv x * v^e$ mod n, and rejects if y = 0.

Steps 3-6 are repeated every time Alice wants to prove her knowledge of the secret.

A → B: x = r2 mod n

A ← B: e ∈ {0,1}

A → B: y = r * s^e mod n

Guillou-Quisquarter(GQ)

The goal of the GQ protocol is to allow A to prove knowledge of S to B in t executions. This is a probabilistic protocol with a probability of v-t for an adversary to fool the verifier. Since the range of possible e values range from 1 to v with v being very large, the probability of an adversary fooling the verifier becomes very small. GQ is an extension of the Fiat-Shamir protocol that depends on the difficulty of factoring[7].

System Parameters

- Trusted center (T) selects RSA-like modulus n=pq, n – public, p and q – secret
- T selects exponent v, v >= 3 and gcd(v, φ) =1 where φ = (p-1)(q-1), s = v-1 mod φ, v – public, s – secret

Per-User Parameters

- Entity A is given unique identity IA

- Redundant identity JA = f(IA) where 1 < JA < n which implies gcd(JA,φ) = 1, f – public function
- T gives A: sA = (JA)-s mod n, sA – secret

Protocol

- A chooses random commitment r, 1 <= r <= n – 1
- A sends B (1): IA, x = rv mod n
- B sends A (2): random e, 1 <= e <= v
- A sends B (3): y = r • sAe mod n

Verification

- B constructs JA = f(IA)
- B computes z = JAe • yv mod n
- B rejects if z = 0
- B accepts if z = x, rejects otherwise

Schnorr

The goal of the Schnorr protocol is to allow A to prove its identity to B. Schnorr is a 3 pass protocol that depends on the difficulty of calculating discrete logarithms. Schnorr's authentication and identification scheme is similar to the above mentioned protocols. It can also be used for digital signatures, when replacing Victor with a one-way hash function[7].

System Parameters

- Select p such that p – 1 is divisible by another prime q (p = 21024, q >= 2160), p, q – public
- Select β, 1 <= β <= p – 1, having order q, α is generator mod p, β = α(p-1)/q mod p, β – public
- Select t, t >= 40, 2t < q

Per-User Parameters

- A chooses secret key a, 0 <= a <= q – 1
- A computes v = β-a mod p

Protocol

- A chooses random commitment r, 1 <= r <= q – 1
- A sends B (1): x = βr mod p
- B sends A (2): random e, 1 <= e <= 2t < q
- A sends B (3): y = a • e + r mod q

Verification

- B computes z = βy • ve mod p
- B accepts if z = x, rejects otherwise

Zero-Knowledge Protocols in Practice

In this part, we consider the practical aspects of zero-knowledge protocols. The presented protocols are still relatively heavy to calculate. Applying these protocols to the real world can be challenging, The use of lighter calculations than public key protocols makes studying this very interesting and very useful.

Real Computational and Memory Requirements

The computational requirements are still heavy for small-scale applications. If the application permits using conventional symmetric algorithms (e.g. DES), those are still greatly lighter and simpler to calculate. There are other security problems in using symmetrical cryptography. If you can only replace one big and very expensive but definitive public key transaction with a series of big and expensive rounds of zero-knowledge transactions, it might not be worth. There is no clear choice for all applications. Especially in small environments, the available computing power and memory is often a limiting factor in the selection of cryptographic techniques. Here is a summary of the requirements of different cryptographic protocol families and their calculation and memory requirements:

Table 1

Protocol Family	Message Size	Protocol Iterations	Amount of Calculation
Zero-knowledge	large	many	large
Public-key	large	one	very large
Symmetric	Small	one	Small

Useful Applications

- Smart-card applications are often mentioned as good places to use zero-knowledge protocols, because they are lighter to compute than the usual public key protocols.
- Useful for electronic cash card, medical information card, intelligent key and lock systems, etc, which should have security to protect the information on access to them.
- Many embedded and most smart card systems could use some degree of real security

Comparison of Protocols

Main Factors used to compare ZK protocols.

- **Communications:** Communications is the number of messages exchanged between the prover and the verifier.
- **Computations:** Computations are the number of modular multiplications for both the prover and verifier. It is the number of on-line and off-line computations.
- **Memory:** The amount of memory used to stored secret keys and other values is another comparison criterion.

- **Security Guarantees**: Security guarantee is the level of security against forgery and disclosure of secret information
- **Trust required in Third Party:** trust is required in a third party. Different trust assumptions can be made between different protocols.

Bandwidth and Memory

GQ allows for a reduction of both memory and bandwidth over Fiat-Shamir.

Security Assumptions

Fiat-Shamir is based on the difficulty of extracting square roots mod n (Factoring). GQ is, also, based on factoring, but this one requires extracting the vth root mod n. Schnorr is based on the difficulty of computing discrete logs mod p. As we have seen in encryption protocols, the bit level for relative security for factoring problem is quite a bit higher than that required of discrete logs[8].

Attacks

Listed below are some of the main attacks used to try to break ZK protocols.

Impersonation: Impersonation is where one entity pretends to be another.

Replay: Replay attack uses an impersonation involving use of information from a single previous protocol execution on the same or different verifier.

Interleaving: An impersonation involving a selective combination of information from one or more previous protocol executions is an interleaving attack.

Reflection: Reflection is an interleaving attack involving sending information from an ongoing protocol execution back to the originator.

Forced Delay: adversary that intercepts a message and relays it later is using a forced delay attack.

Chosen-Text: A chosen-text attack is when an adversary chooses specific challenges in an attempt to gain information about the secret.

Real-Time Applications of Zero-Knowledge Proofs

ZK protocols are used for many real-time applications like authentication, e-voting, watermark verification, etc. Some products like Sky’s Video Crypt, Microsoft’s NGSCB also use ZK protocols[9].

Watermark Verification

In traditional watermark authentication schemes, it is very important to show the presence of watermark in the image without actually revealing it, a prover exposed a watermark to be present in a digital data to the possible dishonest verifier. However, a potential attacker is able to spoil the watermark entirely once classified information like the watermark or the embedding key is known. Some of previous schemes proposed as solutions have not really achieved desirable results. Their lack of security and validity is the most serious problem. we propose a secure watermark verification scheme based on zero knowledge protocols and public-key encryption scheme in order to solve this problem. Water marking scheme, uses zero knowledge interactive proofs based on Digital Signatures to assert ownership on an image. There is no secret information that can be used for removing watermark disclosed during the verification process. This prevents any malicious user from removing the watermark and reselling multiple copies of duplicate watermark.

Sky's Video Crypt

Video Crypt is a cryptographic, smartcard-based conditional access television encryption system that scrambles analogue pay-TV signals. Sky's Video Cryt is an analogue decoding card for satellite descrambler used to authenticate the subscriber's card. it was used initially by Sky TV and subsequently by several other broadcasters.

This uses Fiat-Shamir Zero Knowledge Protocol. The subscriber center holds the public key, secret key and the address while the card holds the public key and address. Every few seconds, the center requests all the cards to authenticate themselves. Each card which is valid has the algorithm for some function F(x) in its ROM while the data for F(x) is in EEPROM. As described earlier in Fiat-Shamir protocol, virtually no knowledge is transferred between F(x) and EEPROM.

Next Generation Secure Computing Base

Next-Generation Secure Computing Base (**NGSCB**), formerly known as **Palladium**, is a software architecture designed by Microsoft. Microsoft's stated aim for NGSCB is to increase the security and privacy of computer users. NGSCB relies on hardware technology designed by members of the Trusted Computing Group (TCG), which provides a number of security-related features, including fast random number generation, a secure cryptographic co-processor, and the ability to hold cryptographic keys in a manner that should make them impossible to retrieve, even to the machine's owner. It was proposed for secure computing environment to use zero knowledge proofing techniques to verify authenticity of services and code. This authentication is called Attestation which uses third party for signature verification. By this way, anonymity of the service is preserved.

Cryptographic Strength: The cryptographic strength of ZKP is based on few hard to solve problems. the one which we have used in our scheme is based on the problem of factoring large numbers that are product of two or more large (hundreds of bits) primes. The values of the public key also changes with every communication, making it more difficult for the attacker to guess it. The prover also generates a random number and the challenge also changes randomly. Thus, with a changed public key, challenge question from verifier and a new random number from the prover, it becomes extremely difficult for the attacker to break the security.

CONCLUSIONS

In this paper, we proposed a new security model to address three important active attacks namely cloning attack, MITM attack and Replay attack. We used the concept of zero knowledge protocol which ensures non-transmission of crucial information between the prover and verifier. Zero-Knowledge protocols allow the prover to prove to the verifier that they know a secret without revealing information about that secret. The proposed model uses social finger print together with ZKP to detect clone attacks and avoid MITM and replay attack. We analyzed various attack scenarios, cryptographic strength and performance of the proposed model. By comparing values between the commitment and response, the verifier can calculate whether the response matches the expected value. This allows the verifier to verify information without having any knowledge of s, the secret to the prover.

REFERENCES

1. Md. Moniruzzaman, Md. Junaid Arafeen, Saugata Bose, Overview of Wireless Sensor Networks: Detection of Cloned Node Using RM, LSN, SET, Bloom filter and AICN Protocol and Comparing

2. Kai Xing Fang, Liu Xiuzhen, Cheng David, H. C. Du, Real- Time Detection of Clone Attacks in Wireless Sensor Networks, Proceedings of the 28th International Conference on Distributed Computing Systems, 2008, Pages 3-10.

3. J. Menezes, P. C. van Oorshot, and S. A Vanstone, Handbook of Applied Cryptography. CRC Press, New York, New York, 1997.

4. W. Trappe and L. Washington, Introduction to Cryptography with Coding Theory. Pearson, Upper Saddle River, NJ, 2nd Edition, 2006.

5. Chor, B., Goldwasser, S., Micali, S. and Awerbuch, B., “Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults”, Proc. of FOCS, pp.383–395 (1985).

6. Amos Fiat, Adi Shamir How To Prove Yourself: Practical Solutions to Identification and Signature Problems

7. Mihir Bellare, Adriana Palacio GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks

8. A. Taleb, Dhiraj K. Pradhan and T. Kocak A Technique to Identify and Substitute Faulty Nodes in Wireless Sensor Networks Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications, 2009, Pages: 346-351

9. Tuyls, Pim T. (Mol, BE), Murray, Bruce (Eastleigh, GB), Efficient Implementation of Zero Knowledge Protocols, United States NXP